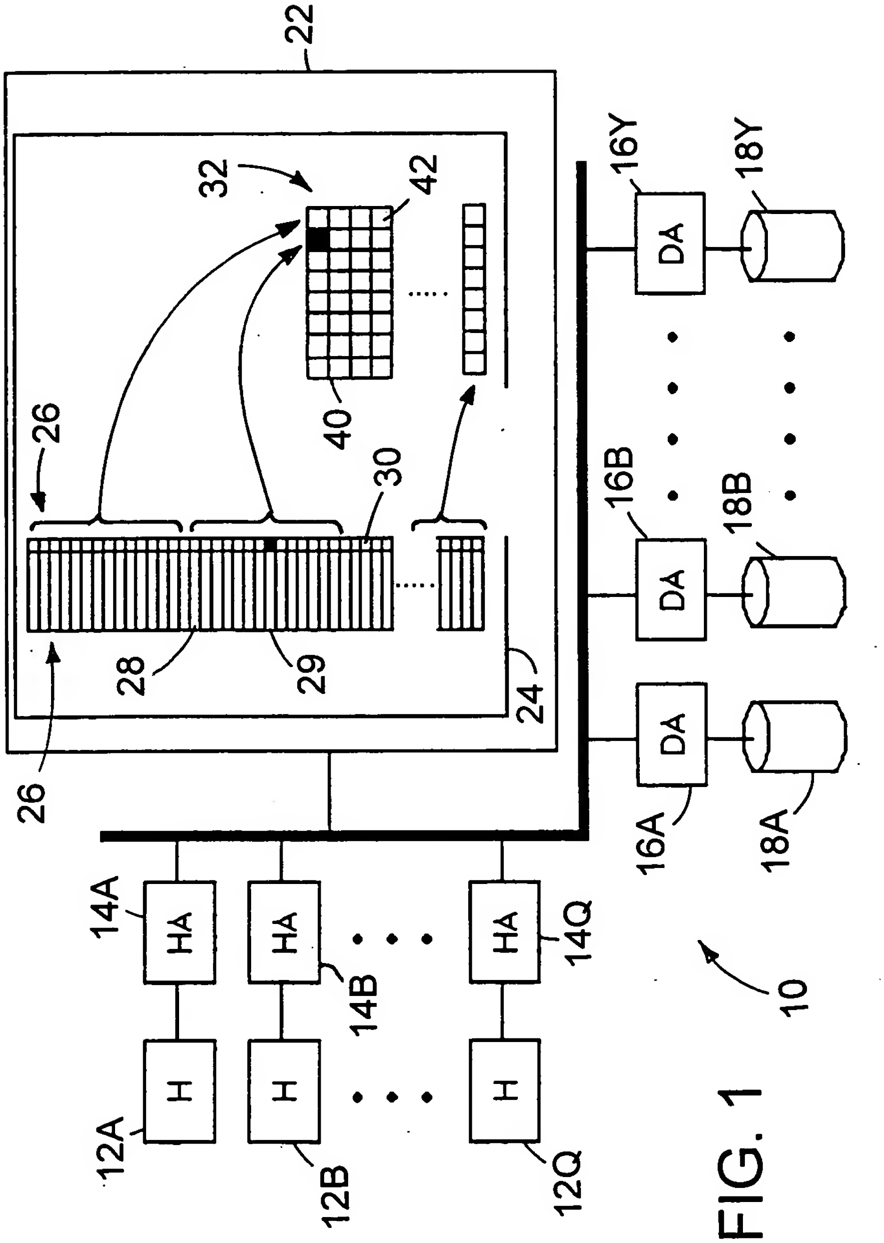
22
24
26
26
28
29
30
32
40
42
H
H
H
12A
12B
12Q
HA
HA
HA
14A
14B
14Q
DA
DA
DA
16A
16B
16Y
18A
18B
18Y
10

FIG. 1

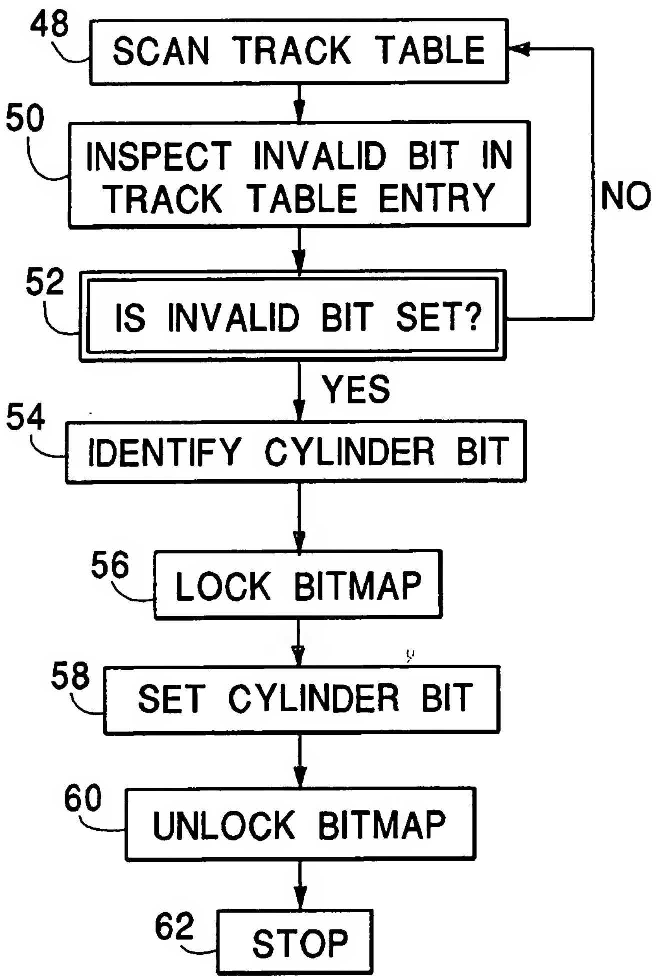
48
SCAN TRACK TABLE
50
INSPECT INVALID BIT IN TRACK TABLE ENTRY
NO
52
IS INVALID BIT SET?
YES
54
IDENTIFY CYLINDER BIT
56
LOCK BITMAP
58
SET CYLINDER BIT
60
UNLOCK BITMAP
62
STOP

FIG. 2

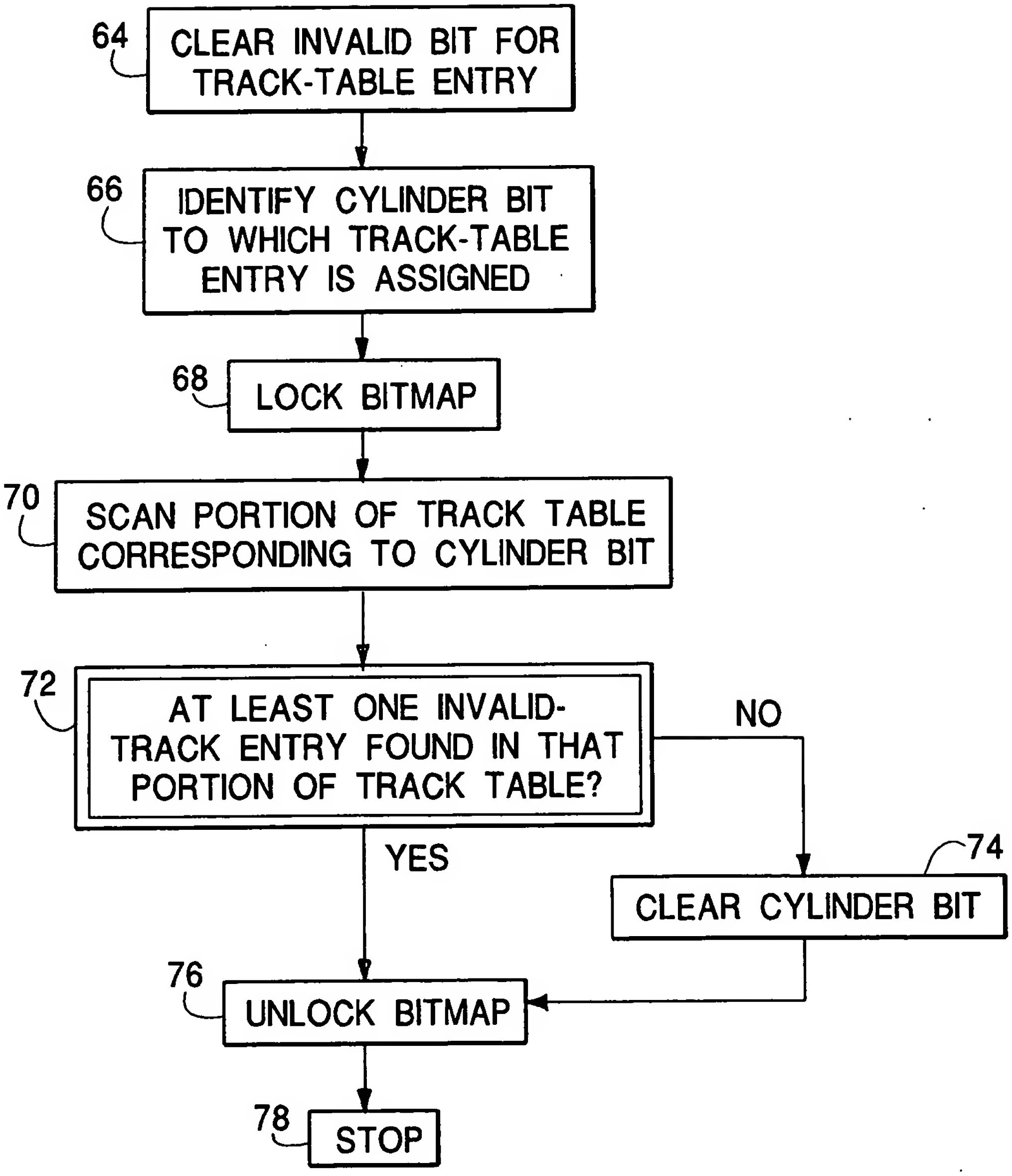
64
CLEAR INVALID BIT FOR TRACK-TABLE ENTRY
66
IDENTIFY CYLINDER BIT TO WHICH TRACK-TABLE ENTRY IS ASSIGNED
68
LOCK BITMAP
70
SCAN PORTION OF TRACK TABLE CORRESPONDING TO CYLINDER BIT
72
AT LEAST ONE INVALID-TRACK ENTRY FOUND IN THAT PORTION OF TRACK TABLE?
NO
YES
74
CLEAR CYLINDER BIT
76
UNLOCK BITMAP
78
STOP

FIG. 3

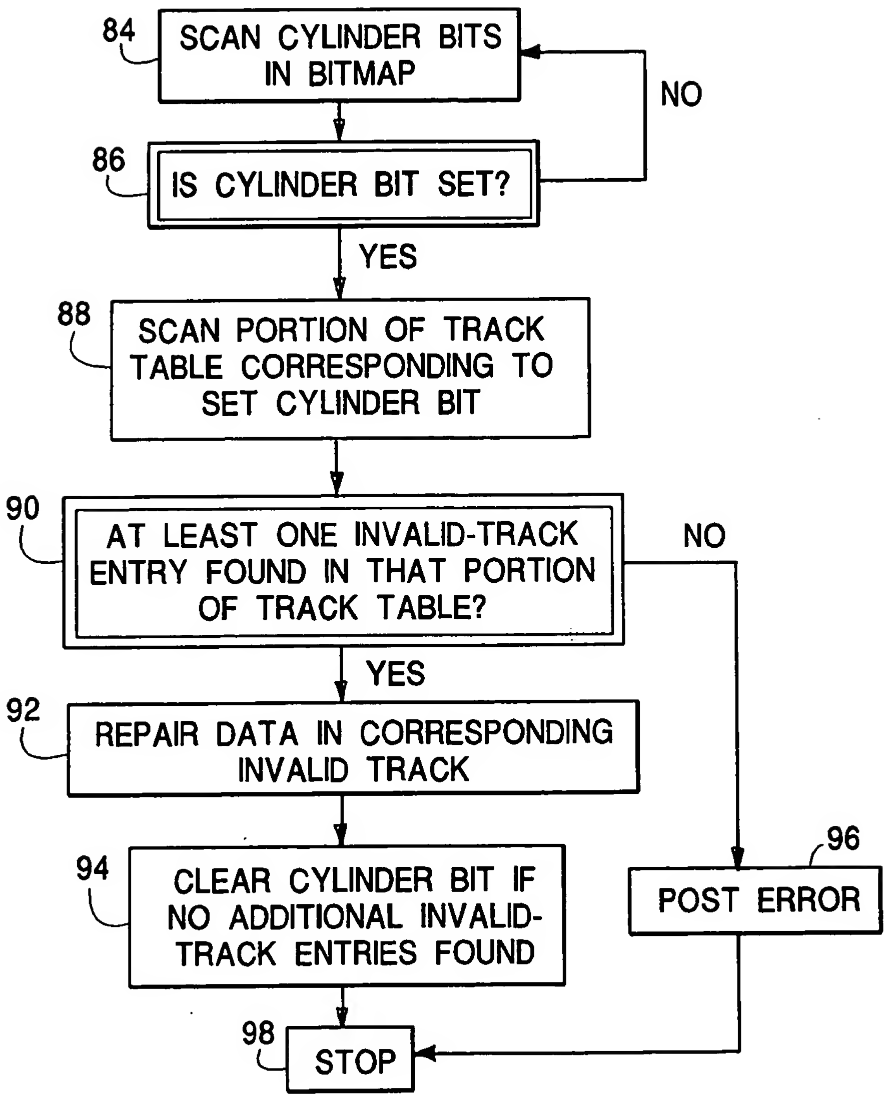
84
SCAN CYLINDER BITS IN BITMAP
86
IS CYLINDER BIT SET?
NO
YES
88
SCAN PORTION OF TRACK TABLE CORRESPONDING TO SET CYLINDER BIT
90
AT LEAST ONE INVALID-TRACK ENTRY FOUND IN THAT PORTION OF TRACK TABLE?
NO
YES
92
REPAIR DATA IN CORRESPONDING INVALID TRACK
94
CLEAR CYLINDER BIT IF NO ADDITIONAL INVALID-TRACK ENTRIES FOUND
96
POST ERROR
98
STOP

FIG. 4

```
      procedure found_invalid_track(T)
      O = (T/15)/8
100   B = remainder(remainder(T/15)/8)
      X = pointer to bitmap
102 — lock bitmap
104 — D = byte at offset O from X
      if (bit B of D) = 0 then
106 ——   set (bit B of D) = 1;
108 ——   write D at offset O from X
      endif
110 — unlock bitmap
```

FIG. 5

```
      procedure fixed_invalid_track(T)
     ┌O = (T/15)/8
112 ─┤B = remainder(remainder(T/15)/8)
     └X = pointer to bitmap
114 ─ lock bitmap
116 ─ D = byte at offset O from X
      if (bit B of D) = 1 then
117 ──── count = 15
         for each track T1 in cylinder containing track T do
118 ────── if T1 is valid then
119 ────────── count=count-1
           endif
         endfor
120 ──── if count=0 then
121 ──────── set (bit B of D) = 0
122 ──────── write D at offset O from X
          endif
       else
124 ──── trace error
       endif
123 ─ unlock bitmap
```

FIG. 6

```
    procedure find_next_invalid_track(T)
    T = last invalid track
    X = pointer to bitmap
126 T1 = remainder(T+1)/n_tracks
    repeat
128   O=(T1/15)/8
      D=byte at offset O from X
130   if D≠0 then
132     for each bit B1 in D do
          if bit B1=1 then
134         for each track T2 in cylinder containing T1 do
136           if T2 is invalid then
                 return T2
              endif
            endfor
          endif
        endfor
138     T1 = remainder((T1+15)/n_tracks)
      else
129     T1=remainder((O*15*8+remainder(T1/15))/n_tracks)
      endif
    until T1=T;
    return -1.
```

FIG. 7